THE UNIVERSITY OF ALBERTA

A COMPUTER TECHNOLOGY GAME - [illegible]

BY

© GUY EVAN RUDY

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF EDUCATION

DEPARTMENT OF INDUSTRIAL AND VOCATIONAL EDUCATION

EDMONTON, ALBERTA

FALL, 1969

UNIVERSITY OF ALBERTA

Faculty of Graduate Studies

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "A Computer Technology Game - Timesharing", submitted by Guy Evan Rudyk in partial fulfilment of the requirements of the degree of Master of Education.

Abstract

This study investigated the possibility of developing, implementing, and evaluating a game to teach computer technology concepts to junior high school students. Computer technology concepts were identified, and integrated into a game which was given the name Timesharing.

Timesharing was revised three times after the initial version was developed. Each revision was a result of criticisms and suggestions made by adults, and students who played the game.

The final version was played by an experimental group of 15 grade 8 students. An equivalent group of students served as the control for the study. The students who participated in the study were selected, and were of higher than average mental ability. From this group of students, the experimental and control groups were randomly assigned.

The experimental group played Timesharing for approximately four hours while the control group continued regular classroom work. Immediately following this, the test instruments were administered.

The test instruments consisted of a Knowledge of (computer technology) Concepts test, and an attitude scale that measured attitude toward the game. The attitude scale was only administered to the experimental group, while the Knowledge of Concepts test was ad-

ministered to the experimental and control groups.

The test items selected for the Knowledge of Concepts test were categorized by two judges, and the judges also scored the test results.

Two hypotheses were tested. The first was that there was no significant difference in learning of computer technology concepts between those students who played the game Timesharing, and those who did not. The basis for acceptance or rejection of this hypothesis was scores obtained on the Knowledge of Concepts test.

The second hypothesis was that there was no significant difference in having a pleasurable experience between those students who play Timesharing, and a hypothetical neutral group. The basis for acceptance or rejection for this hypothesis was scores obtained on the attitude scale.

Both of the hypotheses tested were rejected at the .01 - level of significance.

ACKNOWLEDGMENTS

The investigator wishes to express thanks and appreciation to the following persons for their assistance during the preparation of this study.

To his supervisor, Professor J. E. Gallagher, who provided guidance and encouragement,

To his committee members, Professor M. W. Petruk, and Dr. T. O. Maguire, for their constructive comments, and sincere interest,

To D. W. Manuel and Paul Peel Jr., who participated as judges,

To Marnie Finstad who typed many weekends,

And to his wife, Sharron who helped in many ways.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER I

THE PROBLEM

I. INTRODUCTION

The University of Alberta industrial arts program (Ziel, 1963) recommends two phases of educational experience at the junior high school level. Ziel (1963) describes the first phase as an introduction to tools, machines, and materials, and the second phase as an "elementary introduction to the various technologies prevalent in the world of work (p. 18)."

Computer technology is one of the technologies suggested for inclusion in the second phase of the program, and therefore is included in the industrial arts teacher education program at the University of Alberta. Computers were purchased for the computer technology part of the industrial arts teacher education program.

The curriculum guide for junior high school industrial arts in Alberta, (Department of Education, 1968, p. 62) includes computers as a topic for investigation by students. If computer technology is to be studied by junior high school pupils, in schools which generally do not have computers available for student use, methods of teaching this technology without actually using a computer should be explored.

Various attempts have been made by industry and

education to provide written material and computer simulators to teach computer technology. One such simulator is the Arkay CT 650 desk top computer trainer produced by Comspace Corp. of Brooklyn, New York. The machine is being currently sold by an Edmonton supplier[1] for $625.00. This simulator involves the learner in the control functions of a digital computer by having him actually perform switching operations one step at a time.

Another simulator, not as expensive as the Arkay machine[2], is the Digiac 4030 Digital Computer trainer, manufactured by Digital Electronics Inc. of Plainview, New York. It is useful for teaching programming at a machine language level.

In the past few years, many of the conceptual ideas of mathematics have been made concrete and easily understandable by the development of games in which mathematical relationships were used dynamically. Two such games were Krypto (Yorvich, 1963) which emphasized computation, and Tuf (Brett, 1967) which used equations. If the concreteness and motivational factors of games like these can be applied to teach the concepts of mathematics, perhaps the development of a similar method for teaching computer technology should be explored.

---

[1]. This simulator is handled by Perry School Supplies Ltd., Edmonton

[2]. Quotes received by the Edmonton Public School Board on the Digiac 4030 ranged between $200.00 and $225.00 during 1968.

II. PURPOSE OF THE STUDY

The primary purpose of the study reported here was:

A. To develop a computer technology game to be played by junior high school pupils. Integrated into the game were concepts included in literature related to computers, and in the software of information processing systems.

Secondary purposes of the study were:

B. To identify computer technology concepts that were included in literature related to computers, and as part of the software of information processing systems.

C. To determine student learning of computer technology concepts resulting from experiences playing the game.

III. NEED FOR THE STUDY

The study of computer technology is considered important by educators involved in industrial arts education in Alberta. The Provincial curriculum guide (Department of Education, 1968, p. 62) includes as student experiences and activities in this area the study and use of number systems other than base ten, numeric coding methods for data, and computer instructions. The guide also states that computer languages should be studied, and that students should learn to solve arithmetic problems using a computer. However, no method is suggest-

ed in the guide as to how this unit of learning is to be accomplished.

A possible way to teach computer technology would be to provide students with experiences using a digital computer. Small digital computers, for example the PDP-8/L manufactured by Digital Equipment Corporation, are available for about $10,000.00. It seems unreasonable however, to expect that a school board would provide a small digital computer as part of the equipment for a junior high school industrial arts laboratory, as the recommended approximate total cost of all equipment in a junior high school industrial arts laboratory is less than $20,000.00. (Department of Education, 1967)

Timesharing computer systems could be used to provide a low per terminal cost, and terminals could be placed in a number of schools. Digital Equipment Corporation manufactures a system that provides sixteen terminals, twelve of which have data phone interfacing for remote location use[3]. The per terminal cost is less than $7000.00, an amount that is more than one third of the recommended junior high school industrial arts equipment budget.

The Digiac 4030, and the Arkay CT 650 computer simulators, previously described, are designed to approach

---

[3]. This is an expanded P.D.P.-8/I system, and was marketed during the early part of 1969.

computer system simulation from respectively different perspectives. The Digiac machine is designed to approach computer technology concepts from a programming perspective, while the Arkay machine focuses on computer system organization, and the functions of units within the system.

Information about these simulation systems with respect to their efficiency and effectiveness as aids in teaching computer technology to junior high school students is not made available by the manufacturers. The simulators should be evaluated as to their usefulness for the junior high school industrial arts program.

Computer technology is included as an area of study in the junior high school curriculum for industrial arts. (Department of Education, 1968, p. 62) A variety of perspectives and media should be developed, implemented, and evaluated as curriculum for computer technology in the industrial arts program.

IV. DEFINITION OF TERMS

Computer Technology[4]. Computer technology, is that area of knowledge concerned with the functional organization of information processing systems, methods of encoding and decoding information for these systems,

---

[4]. This definition is not intended to restrict in any way the concepts to be identified in computer technology, but rather to establish parameters within which concepts would be identified.

and operating and controlling these systems.

Game Revision. A game revision is a complete, rebuilt version of the game - Timesharing - in one of its developmental stages. Game revisions are sequentially numbered.

Timesharing. Timesharing is the name given to the game that was developed in the study.

Mnemonic Coding. Mnemonic coding is the process of assigning an alphabetic label to a computer instruction. The label usually provides a cue for identification of the instruction. For example, LDA is the mnemonic for "load the accumulator with the contents of memory address X".

Numeric Coding. Numeric coding is the assigning of a numerical value to a computer instruction. For example, the instruction LDA discussed above, might be assigned the value 12. The encoded value 12 in a computer program would be the signal for the computer to execute the instruction, LDA.

V. HYPOTHESES

1. There will be a significant difference at the .01 level, in learning of computer technology concepts between those students who play the game Timesharing, and those who spend the same period of time in regular class studies.

2. Those students who play the game Timesharing,

will consider the experience pleasurable[5].

VI. DELIMITATIONS OF THE STUDY

The terms significant learning, and pleasurable experience, used in hypotheses one and two respectively, are defined by the test instruments developed for and used in the study.

The computer technology concepts used in the test instrument developed to test the null form of hypothesis one were derived from literature in the area of computer technology, and from computer system software. This topic is discussed in chapter two.

VII. LIMITATIONS

Computer technology concepts are arranged into an organizational structure in chapter three. The structure contains nine concept categories, and each category is represented in the game, Timesharing. No attempt was made to include all computer technology concepts in Timesharing.

The students that participated in the study were not selected at random, and the findings are generalizable to a hypothetical population of similar students. The students that were selected were of above average mental ability as determined by a Lorge Thorndike mental ability test.

---

[5]. The definition of pleasurable is discussed in the delimitations.

CHAPTER II

REVIEW OF RELATED MATERIAL[6]

A distinction is made in this chapter between related literature, which refers to material related to computer technology but not to a specific computer, and computer system software which includes written material designed for a system user. Computer system software is comprised of specific information required to operate a system, and computer programs.

I. RELATED LITERATURE

Computer technology concepts were organized by Robichaud (1966, pp. 107 - 165) into two major areas. These were (a) the computer as a functional system, and (b) information preparation for computer use.

The computer as a functional system had five units; input, storage, central processing unit, output, and a control unit. The control unit regulated inform-

---

[6] A preliminary library search of material available indicated that concepts expressed by a variety of authors varied little, but were organized differently. In effect it became apparent that there were no differences expressed by the authors at a general concept level. In the interests of brevity and conciseness, therefore, the investigator elected to cite those references that were complete in their approach to the subject, rather than exhaustively delineate all the available material. With respect to educational games, the literature described many, but there were no analyses concerning game development, implementation, and evaluation, as curricula.

ation flow throughout the system. Specific devices that acted as each functional unit were discussed and their characteristics described. For example, in the discussion of storage units the relative access time and storage capacity of magnetic core memorys, discs, and drums, were described.

Information encoding techniques, such as magnetic and optical data representation, binary number systems and coding were described by Robichaud in the section concerned with information preparation for computer use. As well, flowcharting and programming were discussed.

Arnold et al., (1966, pp. 179 - 307) essentially covered the same concepts as Robichaud, but in considerably greater detail. The discussion of data representation and binary coding examined a variety of codes, and data flow throughout the system was covered in the discussion of functional units. Flowcharting and programming were covered in some detail, and typical system operation was examined.

A review of computer technology published by Scientific American (1966) discussed coding systems with binary numbers including the concepts of bit and word, and was concerned with logic gating devices and binary arithmetic systems. All the functional units identified earlier were discussed in some detail, as were systems analysis and programming. Timesharing, as a computer system organizational phenomenon was examined. Also, the

use of computers in the areas of science, technology, business, and education was cited.

International Business Machines (1968), in an introductory manual to information processing, identified the functional units of a computer as the central processing unit, storage unit, and input and output units. The central processing unit consisted of two parts - arithmetic/logic, and a control section. Each functional unit was discussed in terms of the actual hardware that performed the function. Binary coding techniques, and encoding to machine readable forms were covered. Systems analysis, programming and generalized system operating techniques were given considerable treatment.

II. COMPUTER SYSTEM SOFTWARE

The user manuals, and related information published by two computer manufacturing companies are discussed in this section.

Fabri-Tek Inc. of Minneapolis markets a computer designed exclusively for teaching computer technology, called the Bi - Tran Six. It is a digital computer with all five functional units, and a memory capacity of 128 six bit words. The system uses a machine level language, and has no provision for using assemblers for higher level languages. Input and output is direct to and from appropriate registers on the computer console, and the coding used is binary coded octal. Peripheral equipment

includes an octal to binary converter, paper tape punch, and a paper tape reader.

The introductory user manual (Fabri - Tek, 1965) began with a discussion of encoding binary numbers, and binary arithmetic. Computer instructions and coding were discussed, and the user was given directions on how to use the computer to do arithmetic. A section was spent on input and output of data, and machine organization was covered. The functional units other than input and output were discussed operationally and instructions were given for the actual operation of these units.

Programming was covered in enough detail for the learner to program the machine in machine language. Each instruction was numerically coded, and the addresses for instructions and operands had to be specified. As well, mnemonic coding was taught. Flowcharting and programming techniques were also covered.

Digital Equipment Corporation manufactures a line of small computers, and in addition to the actual software that is part of these systems, two books that deal specifically with these systems are available to beginner users.

The more general of these two books, Small Computer Handbook, (Digital Equipment, 1967) discussed binary arithmetic and coding, presenting the same system model with functional units that have been discussed previously, and applied this directly to the computer. Mnemonic and numeric instruction coding, and memory

addressing were introduced. Also programming examples with flowcharts were given. After this, applications of the computer, and operation procedures and techniques were covered in detail.

The other publication, Introduction To Programming, (Digital Equipment, 1968) was primarily concerned with programming, but there was discussion of binary coding methods and arithmetic, and of the five unit system model of the computer. Instruction coding and addressing were covered, as was flowcharting, before programming was covered in detail for the computer system. As this company manufactures a time sharing system, a section dealt with features of this system such as the system program library, and user files.

III. SUMMARY

It followed from the literature review that there was an implicit concurrence by most authors that the most general information, and that which a reader must first know in order to understand computer technology, can be divided into three categories.

A. The digital computer is a system consisting of five functional units. The units are input, output, storage, processing, and control. Hardware differences for each of these units results in the units having a variety of specific characteristics. How a unit functions, however, is constant.

B. It is important to have a knowledge of binary coding techniques, binary arithmetic, and encoding methods of changing binary information to a machine readable form.

C. A knowledge of systems analysis, flowcharting, and programming is necessary. Included are concepts of instruction coding, and addressing.

CHAPTER III

PROCEDURES AND RESEARCH DESIGN

This chapter contains definitions of the concept categories used as the basis for the game Timesharing. A description of the development of the game is then followed by a listing of the concepts included in the game. Finally, the test instruments, the sample, and the research design are described.

I. CONCEPT CATEGORIES

The following categories were used to classify the concepts derived in chapter two.

A. The computer as a system. This category includes concepts pertaining to the interrelationships between functional units that comprise a computer system, as well as concepts that define the units. The unit definitions are:

1. Input. This is inputting of information into the computer.

2. Output. This is outputting of information by the computer. The information may or may not be decoded.

3. Storage. This includes all methods of storing information in a computer, and information locating strategies.

4. Control. The control function is that which controls data flow and timing in the system.

5. Processor. This is the unit that performs all logic and arithmetic operations.

B. Input. This category includes concepts pertaining to all the ways and means of inputting information into a computer. The operation of all devices that perform this function such as teletype keyboards, card readers, and magnetic character readers are included here.

C. Output. This category includes concepts pertaining to all the ways and means of obtaining processed information from a computer. Some devices that output information are high speed printers, cathode ray tubes, and digital to analog converters used to perform direct process control functions.

D. Storage. This category includes concepts pertaining to all the ways and means of storing information. The operation of all kinds of memory devices and variables concerned with them such as access time, capacity, and relative cost are included here.

E. Control. This category includes concepts pertaining to the functions of timing and data flow within the computer system. The functions are performed by electronic control systems.

F. Processing. This category includes concepts pertaining to the logic gating that provides the processing facility of a computer. Mathematical algorithms for performing mathematical and logic operations are included.

G. Binary Coding. This category includes concepts per-

taining to binary number systems and their characteristics. Included are conversion from other base systems to base two, and algorithms for doing this.

H. Encoding. This category includes concepts pertaining to coding information to machine readable forms. Encoded information takes the forms of electrical pulses, holes in cards or paper tape, magnetic pulses on magnetic tape, and others.

I. Programming. This category includes concepts pertaining to the preparation of computer programs, and all the ways of organizing information that is to be input into a computer. Systems analysis and flowcharting are included.

II. DEVELOPMENT OF THE GAME

From the concept categories, a game was created that included concepts in as many of the categories as possible. The game was designed for four players, and was given the name Timesharing.

Timesharing, by providing experiences with concrete analogies of abstract concepts, was constructed to teach computer technology concepts to junior high school students. Development proceeded through the successive stages and test sessions described here.

A. The Initial Version

The first version of the game was designed for experimentation with concrete analogs of computer tech-

nology concepts. The hardware was designed to be functional. Visually pleasing graphic design was not given consideration.

The hardware consisted of a game board, (Illustration 1) four sets of 25 word blocks, each set being a different color for each player, (Illustration 2) and a large number of blue and white golf tees, to serve as encoded binary ones and zeros respectively.

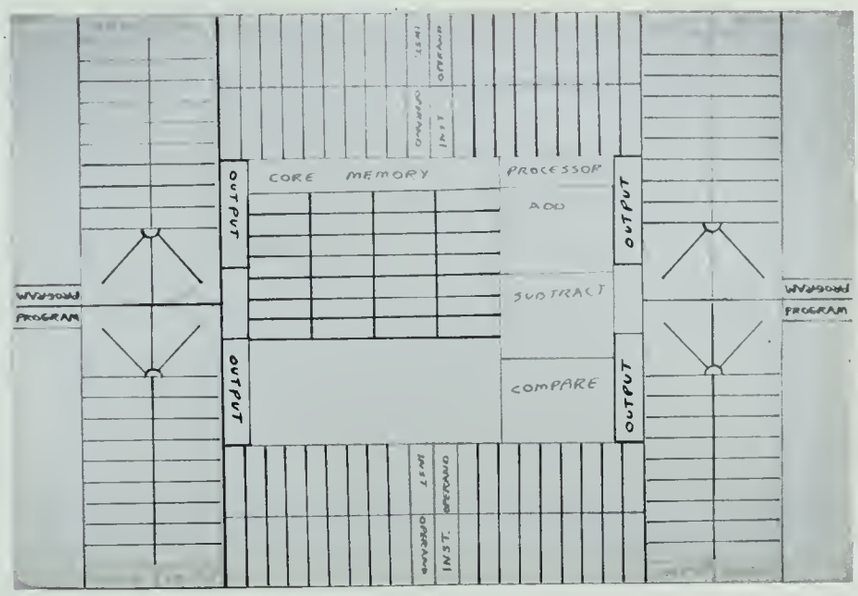


Illustration 1. Game board for the initial version.

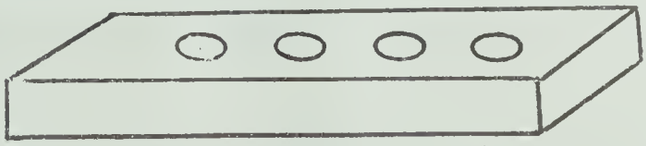


Illustration 2. Word block used in early versions of Timesharing.

The word blocks were to simulate a four bit computer word, and had four holes drilled in them, into which could be inserted four golf tees. All program instructions and data were to be encoded on the word blocks. Within each set of 25 word blocks, 20 were numbered consecutively from one to twenty. These blocks were to contain a player's encoded program instructions. The remaining five word blocks in each set were labelled with a letter, a, b, c, d, and e, and these blocks were to contain a player's data.

The game board was divided, though not labelled, into two main parts, the encoders and the computer. The four encoders were areas on each end where there was a space labelled PROGRAM, and two rows of rectangles with LOAD and INPUT written at the bottom.

Timesharing was played by up to four players, and each player used an encoder. Each player was given a set of rectangular cards which contained all the program instructions necessary to assemble a program to solve a preselected mathematical problem. Each card contained either the operation part of a computer instruction in mnemonic code, or a letter, a, b, c, d, or e, to specify an operand[7].

The cards were shuffled, and each player was given the exact number of cards to assemble the entire program.

---

[7] An operand is the argument contained in a complete computer instruction. For example, in the instruction INPUT A, A is the operand.

A player's program was assembled by arranging the cards in the correct order on the game board in the space labelled PROGRAM. The programming language used was a machine level language similar to that used by the Bi - Tran Six computer. (Fabri - Tek, 1965)

When the program cards were assembled, the players then encoded the program in binary form on a set of word - blocks provided.

Each player received a card with numeric code numbers for each program instruction, and the operands were also given numbers. These numbers were to be converted to binary form, and encoded into four bit words on the word blocks. The word blocks were then arranged, in order, in the rectangles above LOAD and INPUT on the game board to form the encoded program. At this point, the player was ready to start placing his word blocks into the computer.

Each player had a terminal on the game board consisting of an instruction register, and an output register.

To begin using the computer part of the game, each player, in turn, loaded two program words into the instruction register of the terminal. The instruction register consisted of a set of rectangles below INST. and OPERAND on the game board. This part of the game board was "inside" the computer. When an instruction was input into the computer, its two parts, the operator and the

operand were placed in adjacent rectangles under the headings INST. and OPERAND respectively. The complete program was loaded, instruction by instruction, into the instruction register. During any given turn, however, a player was allowed to load only two words. After a player had loaded his words, he said "next!" and the player on his left proceeded. Thus the name Timesharing, as each player used the computer in sequence.

After the entire program was loaded into the rectangles under INST. and OPERAND, a player started using his turns to run his program. Each time his turn came up, he performed one complete operation in his program (one operation consisting of a complete instruction statement including the operand.) For example, if the encoded instruction read "INPUT A", the player would take a word block with the value for A on it, and place the block in CORE MEMORY.

By following the program instructions in sequence, and performing each complete instruction by moving, or changing values on a word block on the game board, the players ran the program by actually performing the instructions.

The entire process was made into a game by the errors that players made. If a player made any error in programming, encoding, or in executing his program, and any other player noticed it, the noticing player said "error!" If it was an error, the player in error lost a

turn, while the noticing player gained a turn. The player who finished executing his entire program first won the game.

The concepts that were included in the initial version, classified into concept categories, were the following.

A. The computer as a system. Each unit of the system was included, and how they were interrelated was demonstrated by acting out the computer instructions.

B. Input. The information was put into the computer by a set, or deck of encoded word blocks, much like a deck of punched cards.

C. Output. Output was not well represented. Output in the game was a still encoded word block, containing the problem solution, in the OUTPUT register.

D. Storage. Storage was a simulated core memory that all players used. It was structured in seven rows and four columns, and could hold 28 four bit words.

E. Control. The control function was inherent in the operations that had to be performed in executing programs; that is, transferring information, each player taking turns, and each unit performing only its function.

F. Processing. The processing was in the form of binary arithmetic that the player had to do at the processor to solve his programmed problem.

G. Binary Coding. Each player had to binary code all the information he used while playing the game. This

included numeric codes for instructions, operands, and data values.

H. Encoding. Each player had to encode all binary information by representing it with different colored golf tees in the word blocks.

I. Programming. Each player used mnemonic and numeric coding of typical machine level language instructions, and assembled the complete instructions into a program.

Evaluation

This version of Timesharing was played four times with adults, and once with teenagers. From these sessions it was recommended that the following changes be included in a revised version of the game.

1. The game board needed an adequate graphic design.

2. There should be repertoire of instruction sheets with the numeric and mnemonic codes.

3. Each player should receive a decimal to binary conversion table.

4. The encoders should be physically separate from the computer game board.

5. More suitable items than golf tees should be found for the bits for the word blocks.

6. The programming and encoding parts of the game were not competitive, and did not add to that aspect of the game. They should be changed.

B. The First Revision

Following the first version of the game that was produced and played, a second version was developed. This second game was functionally the same as the first, but four of the recommendations that were made during play of the first game were incorporated into the second.

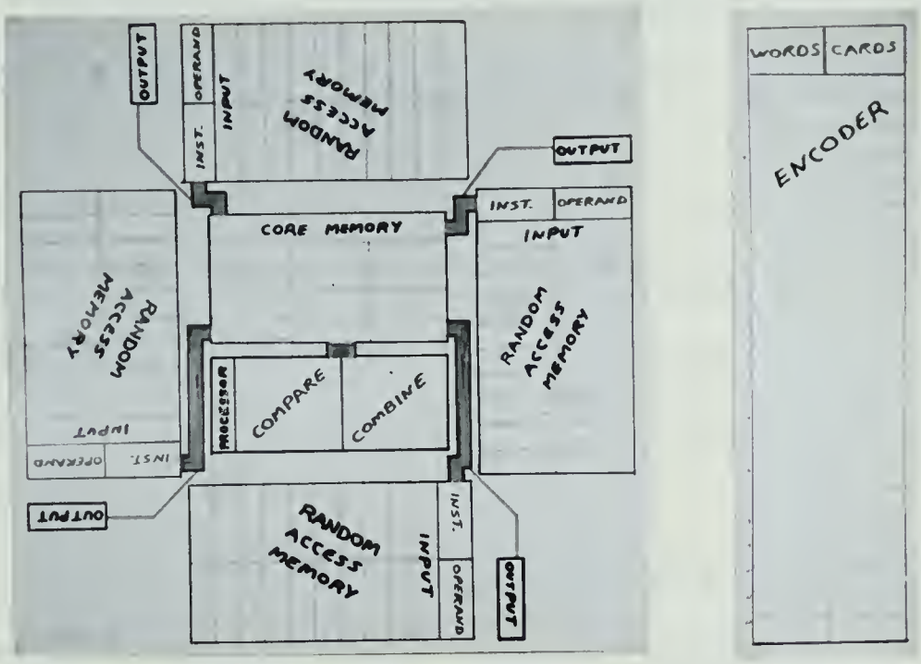


Illustration 3. Game board for the first revision.

These changes included a more pleasing graphic design, and the encoders were made into physically separate units. Also due to these recommendations, sets of rules, lists of program instructions, and decimal to binary conversion tables were included.

Other changes were also incorporated into Time-Sharing. The program storage after input was labelled

as a random access memory, and the processor was changed to have only two operational sections; compare, and combine, while the processor in the first version had add, subtract, and compare.

A set of rules, and a list of program instructions, as well as a decimal to binary conversion table were made available to each player. (See Appendix A and B) As yet, no solution was found to the problem of golf tees for bits, and to the problem of programming and encoding not adding to the competitiveness of the game.

The changing of the processor from one with three units - add, subtract, and compare, to one with two units compare, and combine, was made because the use of add and subtract seemed to indicate that these operations were performed in a separate place in a computer. This is not the case.

The labelling of the storage after input as a random access memory was included to indicate the concept that the program was stored in a memory when it was input into the computer. There was no notion of the concept that the program was stored in a memory after input in the first version of the game.

Evaluation

This new revised game was played twice with adults, one of these times being with a person who had considerable experience in the computer technology field. From

these play sessions the following recommendations ensued.

1. A way must be found to locate any word in the computer system. An addressing scheme was required.

2. In a real system of this type, the core memory would be partitioned among users. Perhaps this feature could be incorporated.

3. The programming and encoding part of Timesharing could be made into a race. Each player would begin at the same time, and whoever got to his computer terminal first could begin inputting ahead of the others.

4. Do not partition the processor at all, but just label it - PROCESSOR.

5. Include program instructions for higher level operations than add and subtract.

6. Use plastic pegs available from toy stores for bits.

7. A control section should be included for each terminal on the game board. In a computer, an instruction must be acquired before it can be executed.

8. The program instruction Start is not a program type of instruction, but is a system command. It should be deleted.

C. The Second Revision

After the second game was played, and recommendations made, the game was again rebuilt. The encoders were left as before, but three significant structural changes were made to the game board.

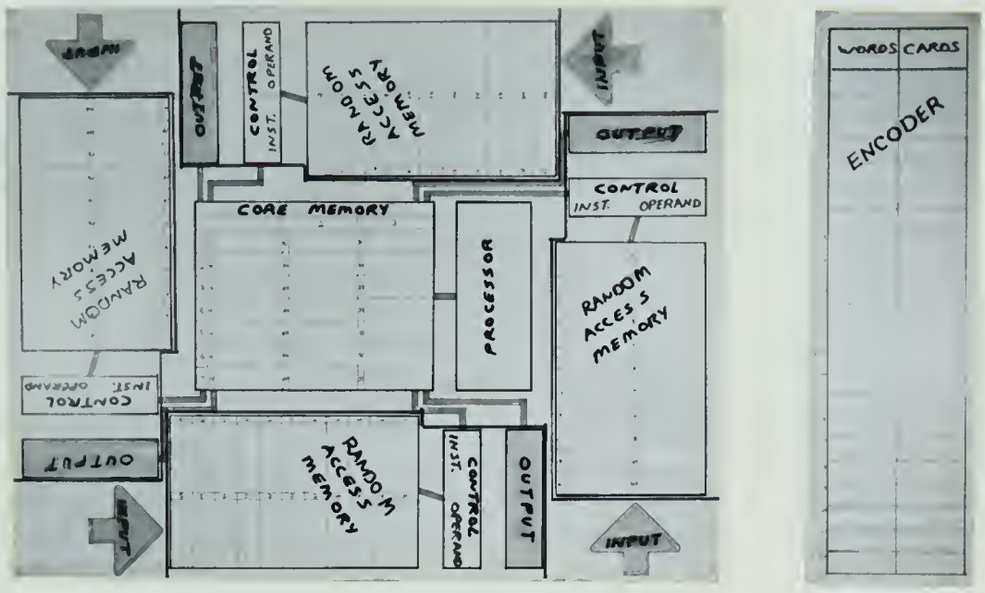


Illustration 4. Game board for the second revision.

The processor was changed so that it was not partitioned at all. The functions of comparing and combining are not carried out at different places in a computer as the old processor seemed to indicate. The new processor was a large rectangle labelled PROCESSOR.

Included at each terminal in the new game was a section labelled CONTROL to contain the instruction to be performed. In play, an instruction and its operand now had to be acquired, and placed into the control section before it could be performed.

Also in this new game, the function INPUT was made clearer by drawing it as a large arrow pointing to the RANDOM ACCESS MEMORY of a player's terminal. This was an

attempt to indicate to the player that he was to "input into" the RANDOM ACCESS MEMORY.

Another hardware change was the use of plastic pegs for bits. These were spray painted either black or white, to indicate ones and zeros respectively.

Functional changes included the addition of multiply and divide to the program instructions, and the program instruction Start was deleted. Now a greater variety of programs was possible due to the multiply and divide instructions.

The programming and encoding part of Timesharing was made into a race, in order to increase the competitiveness of the game.

This version of the game did not include an addressing system, nor was the core partitioned. A satisfactory way to accomplish these two recommended changes had not yet been devised. The addressing shown on the game board in Illustration 4 was not added until after this version of the game was played, and recommendations made.

Evaluation

The game was played twice by adults, one session being played by people with considerable experience in the field of computer technology. The following recommendations followed.

1. In a computer, only one user may be using the processor at a given time. The PROCESSOR should have a

buffer to hold other users information between processing steps.

2. In the programming procedures, an operand address should follow the instruction, not a coded identification for the variable. This could be included when an addressing system is devised.

3. Moving complete instructions to the control section each time a player has a turn is awkward. Perhaps the control area on the board could be placed at the side of each player's RANDOM ACCESS MEMORY, and a moveable arrow could indicate the instruction being performed.

4. It is somewhat difficult to identify the right and left sides of the word - blocks. Perhaps a better method could be found to do this.

5. On the Repertoire of Instructions sheet, each instruction should be carefully defined, as well as numeric and mnemonic codes given.

6. The Rules Of The Game sheet should be made more explicit.

D. The Final Version Of Timesharing

The fourth edition of the game became the final version. This version contained two major changes. The first change was the addition of an addressing system, and core memory partitioning on the game board. (Illustration 5, p. 29) The second change was a re-designed word block.

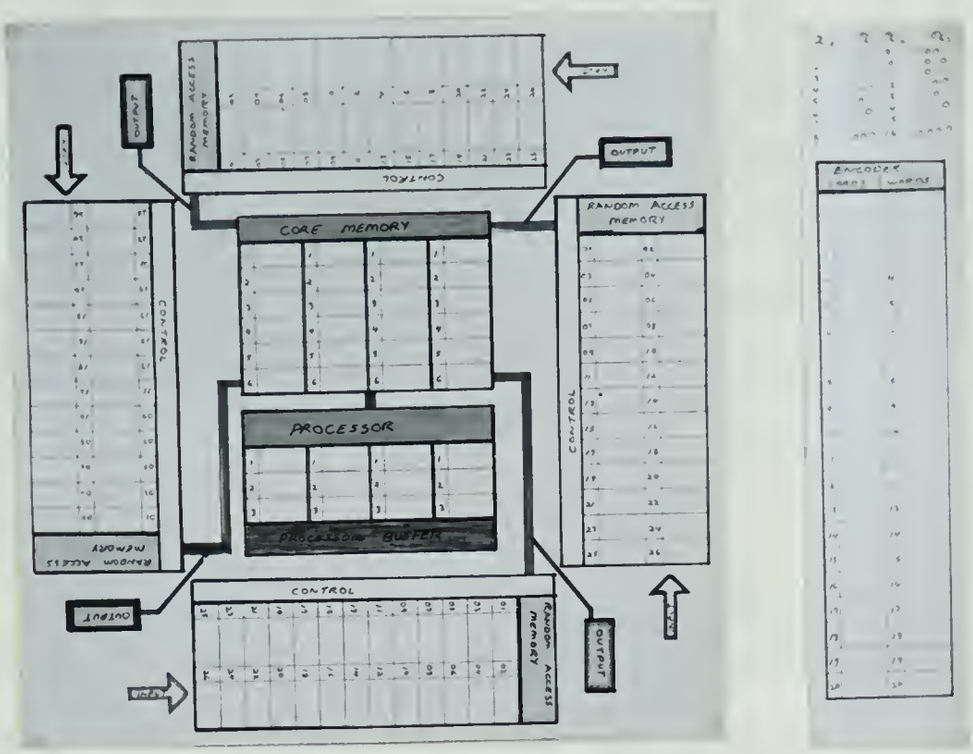


Illustration 5. Game board for the final version.

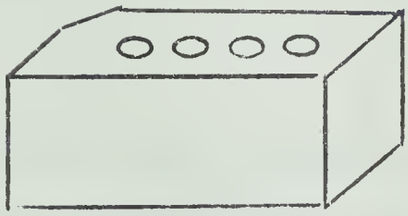


Illustration 6. Revised word block.

The new word - block increased the possible complexity of the game per unit area, by allowing more word blocks to be used in a given area. Also the beveled upper left hand corner on the word block made it easy to identify right and left sides, as well as creating an analogy to

a data processing card.

The addressing system located each word space in the whole game, but the only addresses that are actually used are those in the core memory. This memory had a two axis location system, and was partitioned by color coding into four sections, the colors being the same as the players' word - blocks. As well, the terminals were color coded.

The Repertoire Of Instructions (Appendix B) was changed to include an operational definition of each instruction. Also the rules were expanded, and made more explicit. (Appendix A)

As this final version of Timesharing was used in the experiment, the actual operations required of players as they proceed through play as described here for further clarification.

Prior to play, each player received an ENCODER, a copy of Rules For Timesharing, (Appendix A) and a Repertoire Of Instructions. As well the gameboard was laid out on a table. The game began with each player receiving a set of cards. The set had two kinds of cards; those containing instructions (operators), and those containing operands (arguments). The number of cards in each set was the exact number required to assemble the program for the problem being used to play the game. (For example, the problem a + b = c required a set of fourteen cards, seven being arguments, and seven

being operators). Each player's set of cards for the preselected problem was scrambled, and given to him in an envelope. The player was required to use all of the cards in the set to assemble the program.

At an agreed upon signal, play began. Each player opened his envelope, placed his set of cards on the table, and assembled the program in the space labelled CARDS on the ENCODER. As soon as a player had his set of cards assembled, he began encoding the program on wordblocks, and placing the wordblocks on the ENCODER in the space labelled WORDS.

The numeric coding for the cards containing instructions was given on the Repertoire Of Instructions sheet, and the cards containing operand addresses expressed these addresses numerically. The numerical values for instruction cards, and operand address cards were placed by the player on his own color of word blocks. (A correctly assembled set of cards, and their equivalent word blocks are shown in Illustration 7.)

As soon as a player had all of the word blocks for his program prepared, he prepared word blocks for the data in the problem. (For the problem a + b = c, a player was given data values for a and b). When the players' program, and data were encoded on his word blocks, he returned his set of cards to the envelope, and proceeded to the next stage of the game.

Each player at this stage of the game began loading his program into the computer. He did this by taking

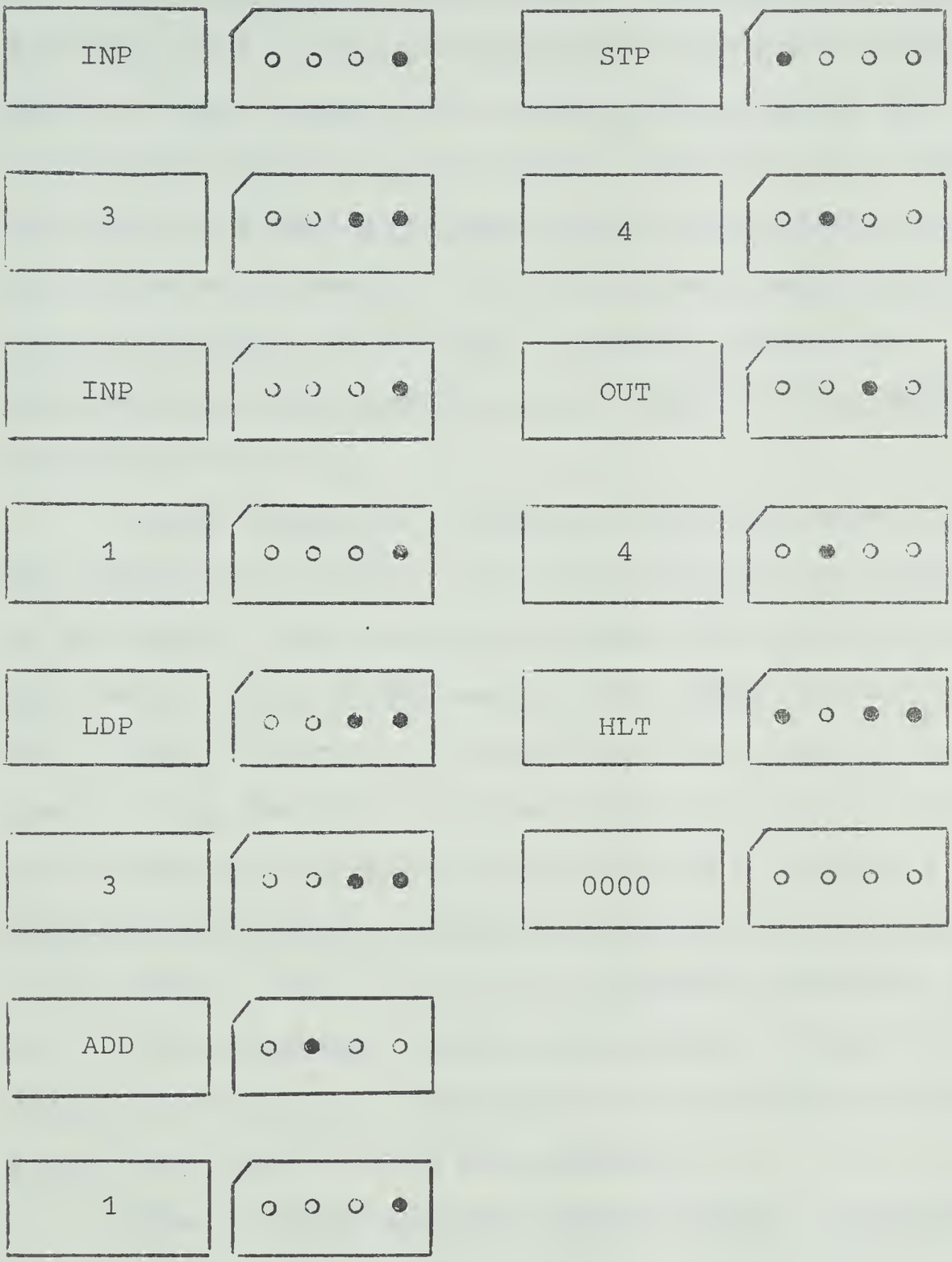


Illustration 7. Assembled Program, And Encoded Word Blocks For A + B = C.

his first two program word blocks, and placing them side by side in the 01 and 02 locations of his RANDOM ACCESS MEMORY. (Each player had a random access memory for holding his program, and an output register as his terminal on the Timesharing game board, color coded to match his set of word blocks). If each player loaded his word blocks correctly, he now had a complete instruction statement including operand in his RANDOM ACCESS MEMORY locations 01 and 02.

After loading his first two words described above, the player said "next!", and play passed to the player on his left. This player then could load two words, and say "next!" Play progressed in this manner to each of the players in turn. If a player was not ready to load words at his terminal, (and he must have had all of his word blocks ready before he could do this), then his terminal was skipped, and play proceeded to the player on his left. (Thus, it was to a players' advantage to be the first to begin loading his program. This often enabled him to have a succession of turns before another player was ready to load his program.)

When a player had his program loaded, he proceeded to "run" it. He did this by placing a small metal arrow in the CONTROL area adjacent to his RANDOM ACCESS MEMORY location 01.

At this point in the game, the player was ready to begin the third stage of play. He proceeded to

execute the instruction indicated on the word blocks in his RANDOM ACCESS MEMORY locations 01 and 02. His word blocks could read "INP 3". This required him to place a word block with the data for "a" on it into CORE MEMORY location 3. (The location 3 that he chose had to be the same color as his word blocks, as the core memory was indexed by number and color).

After a player executed an instruction he said "next!" At this time another player could call "error!" If the player had made a mistake executing his instruction, (for example, transferring data to the wrong location), he lost a turn. The player who called the error gained a turn. If a player called "error!", when in fact there was no error, the calling player lost a turn, and the called player gained an additional turn.

As Timesharing proceeded through successive turns, each player executed his successive instructions by:

(a) moving his arrow down one space.

(b) performing the instruction on the two word blocks indicated by the arrow.

(Operational definitions of each instruction were available to all players on the Repertoire Of Instructions sheet).

The player who completed the total process from sorting his cards, to executing his final instruction and outputting his answer was the winner of the game. Speed in getting to a terminal first, accuracy in

executing his instructions, and accuracy in calling errors on other players, were required factors in winning the game.

Evaluation

This version of Timesharing was played by four grade eight boys for approximately one hour and a half. The boys were purposely selected informally over a range of academic ability, and this investigator spent the entire time with them, providing information as required. The first time through the game was a learning time, and no competition was introduced. The investigator went step by step with the group through one complete game, instructing the students how to play. The problem for programming was a + b = c.

The second time through was a regular game, with the investigator observing unless asked a specific question. The students played through a complete game with few questions, however the boys with least academic ability had some trouble, and were several steps behind at the end of the game, as they were caught making errors several times.

All of the students were asked if they would like to play the game again, and they were enthusiastic about this possibility.

The boys were asked some questions about concepts included in the game, and answered some questions well, but were unsure of others. They all agreed that if they

played the game again, they thought they would be better able to answer the questions. The investigator observed that the boys were having some difficulty in verbalizing what they had learned.

III. CONCEPTS IN THE GAME TIMESHARING

The final version of the game contained the following concepts organized into the concept categories identified previously.

A. The computer as a system. The game contained each of the functional units on the game board. During play of the game, each player performed each function at its location on the board.

B. Input. Each player input his word blocks containing his program into the random access memory of his terminal in a manner analogous to inputting information into a computer from punched cards.

C. Output. The winning player output his result on a word block to the output unit of his terminal. The output information was not decoded.

D. Storage. Storage was a simulated core memory, locations being identified by a two axes system. One axis was a number, the other a color. The colors also served to partition the core, a portion being available to each terminal.

E. Control. The control function was represented by players having successive turns, a control arrow moving

from instruction to instruction in each terminal's CONTROL section, and specific functions being performed in specific places.

F. Processing. This concept was represented by the arithmetic operations that occured at the processor. Also, as a processor can handle only one problem at a time, a buffer was included to store information for incomplete processing operations between player's turns.

G. Binary coding. All information that was to be used in the game was converted from decimal to binary form.

H. Encoding. All information, including program and data had to be encoded onto word blocks, binary values being represented by black and white plastic pegs.

I. Programming. Each player had to organize the cards given him into a program. The language used was a machine level language used by the Bi - Tran Six computer. (Fabri-Tek, 1965) A complete program statement consisted of an instruction, and an operand address. All of the operators have only one argument in this language.

IV. INSTRUMENTS

Knowledge of concepts test. To evaluate learning of concepts included in the game, a test was constructed using the following parameters. (See Appendix C)

1. Each concept category was represented by two questions, one directly related to the game, and the other a generalized question.

2. The test was validated by two judges who have graduate degrees in education, and have had experience in the field of computer technology. The judges were given a list of the concept categories, along with their descriptions, and were asked to sort the test items into appropriate categories. The questions used were placed into the categories independently, and unanimously by both judges.

3. The test was given to the experimental and the control groups [illegible] in the study.

Scoring. The test was scored by the two judges who validated the test. The tests were administered to the group of four students who took [illegible] prior. They and the results were [illegible] to [illegible] the judges. [illegible] discussion, the [illegible] might [illegible] preferred to score the test [illegible] independently, as some of the answers seemed [illegible] a need for discussion before they were given a mark. The test papers for the experimental and control groups were not separated, and no identification of the group [illegible] on the papers when they were given to the judges for scoring.

Statistic. The statistic used to test acceptance or rejection of the null form of hypothesis one was Student's t (Ferguson, 1959). The [illegible] statistic was used to determine the relationship between the results of the experimental and control groups with respect to game specific, and general questions, as separate categories.

Attitude Questionnaire. A questionnaire was constructed using a "Likert - type scale" (Seltiz et al., 1965, p. 366) to measure student attitude toward the game, and was administered to the experimental group. The questionnaire is illustrated in Appendix C. The respondents were required to respond in terms of five categories of agreement or disagreement.

Numerical values assigned to items ranged from one point for checking strongly disagree, to five points for checking strongly agree for positive items. Reverse scoring was used for negative items, and there were an equal number of positive and negative items.

V. THE SAMPLE

The sample was selected from two grade eight classes in a junior high school in the Edmonton Public School District. Thirty two students were selected, sixteen from each class. The thirty two were randomly divided into two groups using a randomized block design, the blocking variable being mental ability. Students chosen for the sample had scored from 100 to 145 with a mean of 120 on the verbal scale, and from 105 to 139 with a mean of 119 on the non verbal scale of the Lorge Thorndike test of mental ability. The students lived in a new urban subdivision, and this was the first year of operation of their school.

VI. RESEARCH DESIGN

The research design took the form of two randomly selected groups, one becoming the experimental group and the other the control group. The control group members were aware that they were part of the investigation before the experimental group played Timesharing. The control group was told that they would have the opportunity to play Timesharing when the experiment was completed. Some students from the control group did play Timesharing later.

During the time that the experimental group was playing Timesharing, the control group students were engaged in regular classroom activities. Neither the experimental, nor the control group had any previous experience in the area of computer technology prior to this investigation. After the experimental group accumulated approximately four hours experience with the game, the test instruments were administered to both the experimental and control groups.

R X O

R O

Illustration 8. Research design

Timesharing was played for about two hours in the morning, and two in the afternoon by the experimental group. Also the experimental group was divided into two

groups of eight each, as only two copies of Timesharing were available, and each game could accommodate four players. The experiment, therefore took two days to complete. The days were Monday and Friday, and were determined by the timetable of the school. The experiment took place following the Easter holidays.

The test instruments were administered immediately following the afternoon play sessions.

CHAPTER IV

RESULTS OF THE STUDY

The results of the investigation are presented in six sections. The first section discusses the concept categories and concepts derived in chapter two. The second section tabulates the raw scores resulting from the knowledge of concepts test, the attitude scale, and the verbal and non-verbal mental ability tests. The third section compares the experimental and control groups by scores obtained on the knowledge of concepts test. Fourth, is a report of the experimental groups' attitude towards the game, and fifth, the results obtained by girls as a separate group, within the experimental and control groups, are examined. The sixth section examines specific items from the knowledge of concepts test.

I. COMPUTER TECHNOLOGY CONCEPTS

From the literature, and computer system software review, concepts were organized into the following three categories.

A. The functional organization of a computer consisting of input, output, storage, processing, and control units. This category includes each unit separately, and how the units function together as a

system.

B. Binary coding techniques, binary arithmetic, and encoding methods of changing binary information to a machine readable form.

C. Systems analysis, flowcharting, programming methods and techniques.

These three categories were expanded into the nine concept categories used as the content basis for Timesharing.

II. RAW SCORES

The raw scores obtained by the experimental group are displayed in Table I, and those obtained by the control group in Table II. The sum, arithmetic mean, and standard deviation are included for each test administered to each group.

III. KNOWLEDGE OF CONCEPTS TEST SCORES

The results obtained by the experimental and control groups were examined in three ways. First, a t statistic was used to compare means for total scores attained, and scores on general questions attained, by the experimental and control groups on the knowledge of concepts test. Second, Spearman's coefficient of rank correlation was used to determine the relationship between the rank order of total scores, and scores on

general questions for both the experimental and the control groups. Third, for the experimental and control groups, scores on the total test, and on the generalized questions, were stated as proportions of the totals possible.

For testing purposes, hypothesis one was stated in the null form.

Hypothesis one. There will be no significant difference at the .01 level, in learning of computer technology concepts between those students who play the game Timesharing, and those who do not. (Rejected.)

For the mean obtained on the knowledge of concepts test by the experimental group, compared to the mean obtained by the control group, t = 10.97. At a .01 level of significance, with 29 d.f., the t - value must be less than -2.76 or greater than 2.76. The null hypothesis therefore, was rejected.

The means obtained by the experimental, and control groups on the generalized questions, on the knowledge of concepts test, were also compared. For these two sets of scores t = 5.73. At a .01 level of significance, with 29 d.f., the t - value must be less than -2.76, or greater than 2.76. Therefore the null hypothesis one could also be rejected on the basis of a comparison of the generalized question scores of the experimental and control groups. Spearman's formula for

TABLE I

RAW SCORES ON THE EXPERIMENTAL GROUP ON THE MENTAL ABILITY TEST, KNOWLEDGE OF CONCEPTS TEST, AND THE ATTITUDE SCALE

N	V	NV	T_t	T_g	A
1	131	127	31	14	45
2	134	120	61	33	47
3	110	116	63	29	48
4	117	111	51	23	47
5	145	119	56	20	47
6	113	119	40	20	49
7	126	108	54	29	41
8	105	139	36	22	37
9	136	117	53	24	41
10	119	118	44	19	41
11	110	129	39	11	44
12	117	119	47	20	47
13	120	112	35	15	45
14	112	118	34	16	33
15	117	109	52	24	43
total	1812	1781	696	319	655
mean	120	119	46.4	21.3	43.7
S.D.	10.95	7.81	9.88	5.84	4.28

N - subject's number
V - score obtained on verbal section of Lorge Thorndike test of mental ability
NV- score obtained on non-verbal section of Lorge Thorndike test of mental ability
T_t- score obtained on knowledge of concepts test
T_g- score obtained on generalized questions on knowledge of concepts test
A - score obtained on attitude scale
S.D.- standard deviation

TABLE II

RAW SCORES OF THE CONTROL GROUP ON THE MENTAL ABILITY TEST, KNOWLEDGE OF CONCEPTS TEST, AND THE ATTITUDE SCALE

N	V	NV	T_t	T_g
1	120	105	17	12
2	110	118	8	8
3	112	110	9	9
4	126	105	11	11
5	117	119	9	9
6	123	122	20	16
7	120	125	8	8
8	124	113	8	8
9	124	114	5	5
10	124	116	20	12
11	129	126	4	2
12	100	123	19	15
13	119	114	13	13
14	113	121	23	16
15	133	113	5	1
16	125	131	23	17
total	1919	1875	202	162
mean	119	117	12.6	10.1
standard deviation	7.93	7.12	6.46	4.66

N - subject's number
V - score obtained on verbal section of Lorge Thorndike test of mental ability
NV- score obtained on non-verbal section of Lorge Thorndike test of mental ability
T_t- score obtained on knowledge of concepts test
T_g- score obtained on generalized questions on knowledge of concepts test

rank correlation was used to determine the relationship between rank order of answers on the generalized questions, and rank order of answers on the total questions, for both experimental and control groups.

The r obtained for the experimental group's results was .827, indicating a high correlation between the rank order on the total, and generalized question results, for the experimental group.

The r obtained for the control group with respect to total questions, and generalized questions on the knowledge of concepts test was .877, indicating a high rank order correlation.

To determine proportionate differences between total scores, and generalized question scores, the mean of total scores, and the mean of generalized question scores, were stated as a fraction of the total score possible on the test, and the total possible on the generalized questions respectively. This was done for both the experimental and control groups. (See Table III)

IV. ATTITUDES OF STUDENTS

The attitude scale was administered to the experimental group. A neutral attitude towards the game would yield a score of 30 on the scale. As there were 15 students in the experimental group, their scores as a group were compared to a hypothetical group having a

TABLE III

PROPORTIONATE DIFFERENCES BETWEEN TOTAL SCORES, AND GENERALIZED QUESTION SCORES, OF THE EXPERIMENTAL AND CONTROL GROUPS

	T_t	$\bar{X}_t$	$\bar{X}_t/T_t$	T_g	$\bar{X}_g$	$\bar{X}_g/T_g$
Experimental Group	74	46.4	.628	40	21.3	.532
Control Group	74	12.6	.170	40	10.1	.253

T_t - total possible, total questions

$\bar{X}_t$ - mean obtained, total questions

T_g - total possible, generalized questions

$\bar{X}_g$ - mean obtained, generalized questions

binomial distribution of scores on the attitude scale. The hypothetical group was given a n of 15, and a mean of 30.

Hypothesis two was stated in a null form for testing.

Hypothesis two. There will be no significant difference at the .01 level, in having a pleasurable experience, between those students who play Timesharing, and a hypothetical neutral group. (Rejected.)

A t - test was used to compare the means obtained by the two groups, and a t score of 28.83 was the result. At a .01 level of significance, with 29 d.f., the t - value must be less than -2.76 or greater than 2.76. The null hypothesis, therefore, was rejected.

Spearman's formula for rank correlation was used to determine the relationship between rank order of answers correct on the attitude scale, compared to rank order of answers correct on the knowledge of concepts test, for the experimental group. The r obtained was .389, indicating a low correlation.

V. SCORES OBTAINED BY GIRLS

There were five girls in the experimental group, and seven in the control group. The girls' scores were compared to those of the boys' in Table IV. The girls in the experimental group obtained scores on the know-

TABLE IV

SCORES OBTAINED BY BOYS AND GIRLS ON THE MENTAL ABILITY TEST, KNOWLEDGE OF CONCEPTS TEST, AND THE ATTITUDE SCALE

	N	V	NV	T_t	T_g	A
Means - experimental group						
girls	5	127	119	52.4	23.8	46.8
boys	10	118	119	43.4	20.0	42.1
Means - control group						
girls	7	118	115	11.7	10.4	
boys	9	121	119	13.3	9.0	

V - score obtained on verbal section of Lorge Thorndike test of mental ability
NV - score obtained on non-verbal section of Lorge Thorndike test of mental ability
T_t - score obtained on knowledge of concepts test
T_g - score obtained on generalized questions on knowledge of concepts test
A - score obtained on attitude scale
N - number of students in group

ledge of concepts test, and on the attitude scale that were somewhat higher than those of the boys. The mean verbal mental ability score for the girls was also higher than the mean for the boys.

VI. KNOWLEDGE OF CONCEPTS TEST ITEMS

The specific items included here are reported because they were answered in particularly interesting ways. Item two, a generalized item, read: "A computer has five main parts. What are these?" The answer is input, output, control, processor, and memory. In the experimental group, nine students included in their answers two types of memory; core memory, and random access memory, and left out one of the functional units. Nine students included processor buffer as a main part, and five included encoder.

Question four required the students to convert a binary number to a decimal number. This item was answered incorrectly by seven students in the experimental group. Question three, which required students to convert a decimal number to a binary number, was answered incorrectly by one student in the experimental group.

Question twelve read: "How would information on I.B.M. cards get into a computer?" Five students in the experimental group indicated that the card itself is fed into the computer, and two students stated that the cards

are actually transferred around inside the computer, eventually arriving at the output.

Test item two asking for the five functional units of a computer was answered more often correctly by students in the control group than other questions. Five students listed input, nine listed output, six listed memory, two listed processor, and three listed control.

Also item eight, asking how an I.B.M. card contains information was answered correctly by students in the control group more often than other questions. On this item, they accumulated a total of 32 points out of a possible total of 64.

CHAPTER V

DISCUSSION AND IMPLICATIONS FOR FUTURE RESEARCH

In review, the major objective of this study was to develop a game that would teach computer technology concepts to junior high school students and at the same time be pleasurable to play. How well the teaching of computer technology concepts occurred, was measured by a test that was validated and scored by two judges. The pleasurable aspect was determined by an attitude scale administered to the experimental group who played the game.

The game proceeded through several stages of development, being played by, and criticisms being considered from, teenagers, adults, and adults who had experience in the field of computer technology.

I. DISCUSSION

The findings show that the objectives of this study were achieved. It is possible to teach computer technology concepts to junior high school students with the game Timesharing. Also it is highly likely that students who play the game will find the experience enjoyable. It must be cautioned however, that the research design does not permit generalization to the population from which the students that comprised the

experimental and control groups were selected. The t statistic used for hypotheses one and two permits inference of the results of the experimental group to a hypothetical similar population.

It is interesting to note that the experimental group scored an average proportion of .628 of questions correctly on the knowledge of concepts test. With respect to generalized questions, the proportion was .532 correct of the total possible. Students in the experimental group scored nearly as high proportions on generalized questions as on total questions.

Also there was a rank order correlation of .827 between total test responses, and generalized question responses of students in the experimental group. Therefore students were likely to obtain similar rank order of scores on total questions and generalized questions. A suggested explanation for this, is concepts learned from playing the game become generalized and are transferrable.

This explanation is reinforced by an examination of the way in which some questions were answered on the knowledge of concepts test. Item two, which read: "A computer has five main parts. What are these?", quite often received core memory, and random access memory as answers for seperate functional units. Also processor buffer, and encoder were included quite often as answers. All of these answers existed as sections on the Time-

sharing gameboard, and were labelled core memory, random access memory, processor buffer, and encoder.

Another generalized item on the knowledge of concepts test, asked the students how information on an I.B.M. card gets into a computer, and was answered in a way that reflected what the students were required to do while playing the game. In the game, the actual encoded word blocks are moved around from unit to unit on the gameboard. The student responses to the question indicated that cards are actually input into a computer, and moved around inside.

The control group scored an average proportion of .170 of questions right on the knowledge of concepts test, while scoring an average proportion of .253 right of the generalized questions. These students obtained proportionate scores on the generalized questions that were nearly 1.5 times greater than their proportionate scores on the total test.

It was expected that students who had not played the game would not fare well on the game specific questions. What was interesting is that they scored as well as they did. A possible explanation is that the game specific questions were general enough for students to apply knowledge they had gained in their reading activities. Articles that discuss computer technology concepts are popular in newspapers and magazines. Also there

were a number of books that discussed computer technology in the school library. The rank order correlation between total question scores, and generalized question scores for the control group was .877, reinforcing this explanation.

The attitude of students toward the game was indicated as positive by the attitude scale. The most negative attitude toward the game would be indicated by a score of 10, and the most positive attitude would be indicated by a score of 50. The mean score of the experimental group of students was 43.7, with a standard deviation of 4.28. The t test used to determine acceptance or rejection of hypothesis two assumed that a group the same size as the experimental group, with a neutral attitude towards the game, could be represented by a binomial distribution with a mean score of thirty on the attitude scale.

The space at the bottom of the attitude scale sheet where students were invited to comment on the game, was responded to by 14 of the 15 experimental group students. Of these 14 students, 11 responded with a positive statement about the game. Two students made suggestions on the construction of the game, and one said he would like to see "a real computer operate this way automatically." Three students indicated that Time-sharing was more fun than the mathematical games avail-

able in the math laboratory in the school, and three hoped the game would be available in their school next year. There were no negative comments about the game.

The scores obtained by the girls on the knowledge of concepts test were somewhat higher than those of the boys. There are two factors that may contribute to a possible explanation for this. The first is that the verbal mental ability score for the girls in the experimental group was higher than that for the boys. This occurred in spite of the fact that when students were selected to take part in the investigation, sex was not a variable that was considered in the selection. The second is that the game sessions were conducted in an instructional media centre, in the industrial arts laboratory, in the school. This environment was totally new to the girls, as was the investigator as a teacher. The environment and the investigator were familiar to the boys.

II. IMPLICATIONS FOR INDUSTRIAL ARTS EDUCATION

Although the study has demonstrated that computer technology concepts can be learned by playing the game Timesharing, further research is required with respect to this media for teaching computer technology concepts.

The sample used for this study was small, and selected. The study should be replicated with a much

larger experimental group randomly selected from a large population. Industrial arts has available computer simulators and written material to teach computer technology. These should be compared to Timesharing in terms of learning achieved per unit time for students, and teacher time required. Also a similar study could compare the use of actual computers, computer simulators, and Timesharing to teach computer technology. For studies of this type, relative financial cost should also be considered.

The game, Timesharing, might be used as an introductory experience in the computer technology area, followed by experience with simulators, or computers. The game could thus be evaluated as an introductory experience.

With respect to the game itself, areas for further research are indicated. The graphic design of the gameboard was reflected in the answers to generalized questions on the knowledge of concepts test, as were the actual things the players had to do while playing the game. When asked to list the main parts of a computer, students tended to list items that were displayed dominantly on the gameboard. Also, while playing the game, students were required to physically move quantities of encoded data around. In answering test questions, some students indicated that they thought data processing

cards were actually moved around inside a computer.

In both cases above, students gained misconceptions by playing the game. Perhaps the game could be revised in light of this.

Timesharing, as a means to teach concepts in computer technology, is feasible for use in junior high school industrial arts programs. Its material cost would be low compared to the cost of simulators now available, and compared to the cost of computers. Further research is required, however, to determine the efficiency and the effectiveness of Timesharing as a teaching media.

B I B L I O G R A P H Y

BIBLIOGRAPHY

Arnold, R. R., Hill, H. C. and Nichols, A. V. Introduction to Data Processing. New York: John Wiley, 1966.

Canadian Education Association Research and Information Division. Just What is Data Processing? Information Bulletin. Toronto: 1967.

Department of Education. Alberta Industrial Arts Equipment List. Edmonton: Province of Alberta, 1967.

Department of Education. Junior High School Curriculum Guide for Industrial Arts. Edmonton: Province of Alberta, 1968.

Digital Electronics, Inc., Digiac 4030 Programming Training Manual. Plainview, New York: DEI, 1965.

Digital Equipment Corporation. The Digital Small Computer Handbook. Maynard, Massachusetts: DEC, 1967.

Digital Equipment Corportation. Introduction to Programming. Maynard, Massachusetts: DEC, 1968.

Fabri - Tek, Inc. Bi - Tran Six: Introduction to Computer Concepts Workbook. Hopkins, Minnesota: FTI, 1965.

Ferguson, G. A. Statistical Analysis in Psychology and Education. Toronto: McGraw - Hill, 1959.

Gibson, D. E. An Introduction to Automated Data Processing. Elmhurst, Illinois: Business Press, 1966.

International Business Machines Corporation. Introduction to IBM Data Processing Systems. White Plains, New York: IBM, 1968.

International Business Machines Corporation. Principles of Programming. White Plains, New York: IBM, 1961

Oakford, R. V. Introduction to Electronic Data Processing Equipment: Its Operation and Control. Toronto: McGraw - Hill, 1962.

Robichaud, B. Understanding Modern Business Data Processing. Toronto: McGraw - Hill, 1966.

Scientific American, Inc. Information. San Francisco: W. H. Freeman, 1966.

Selltiz, C., Jahoda, M., Deutsch, M., and Cook, S. W., Research Methods In Social Relations. (Rev. Ed.) New York: Holt, Rinehart, and Winston, 1965.

Ziel, H. R., A new concept of the industrial arts program. The ATA Magazine, 1963, 43 (6), 17 - 21

A P P E N D I X A

RULES FOR TIMESHARING

RULES FOR THE FIRST AND SECOND REVISION

RULES FOR TIME-SHARING

1. The object of the game is to:
   a prepare a program
   b encode the program
   c "run" the program

2. At level one, the program is pre - selected.

3. To start the game, each player is given a set of cards and an ENCODER. The cards are to be arranged into a program under the heading CARDS on the ENCODER.

4. All cards must be used.

5. All players must begin at the same time.

6. When programming is finished, the players must encode the program in binary form on the WORD BLOCKS provided.

7. A white tee = 0. A blue tee = 1.

8. As soon as a player has encoded his program, he must remove the program cards, and proceed with the encoded words.

9. At this point the player is ready to load his program into the RANDOM ACCESS MEMORY of the computer. The pulse cycle is two, therefore the player may only load two words.

10. After loading two words, he must say "next!" and then the player on the right may load two words. If this player is not ready, however, play passes to the next player, etc.

11. If no other player is ready, the first player again loads two words, and says "next!"

12. When all players get information into the computer, each player may only move two words when it is his turn.

13. Any player who makes an error in programming, encoding, or following his own program through the computer looses a turn.

14. A player who discovers any other player making an error gains a turn.

15. The player who gets his program through the computer first wins.

RULES FOR THE FINAL VERSION

RULES FOR TIMESHARING level one

1. The object of the game is to:
   a. prepare the program
   b. encode the program
   c. "run" the program on the game board that simulates a computer.

2. At level one, the program is pre - selected.

3. To begin the game, each player is given a set of cards and an ENCODER. The cards are to be arranged into a program under the heading CARDS on the ENCODER.

4. All the cards in the given set must be used in assembling the program.

5. The program instructions must follow a sequence of:
   a. input data
   b. process data
   c. output result
   d. stop

6. When assembling the program, each instruction card must be followed by an operand address card. The operand address card gives the location in CORE MEMORY of the quantity the instruction is to be done

to. For example the cards

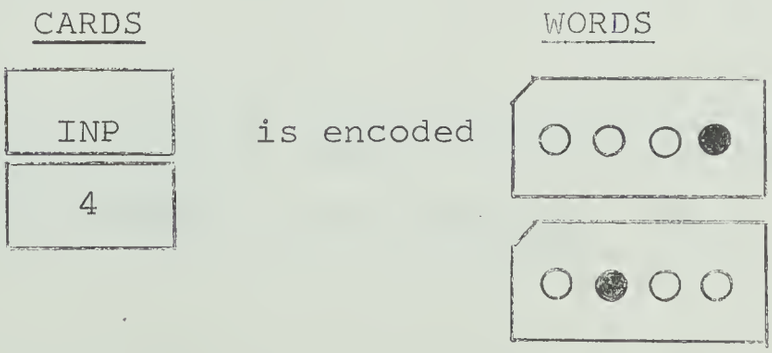
in this order mean "place a quantity in CORE MEMORY location 3." Similarly the cards

in this order mean "add the contents of memory location 5 to any quantity in the processor.

7. All players must start the game at the same time.

8. When programming is finished, the players must encode the program in binary form on the WORD BLOCKS provided. Each instruction is encoded according to its number on the REPERTOIRE OF INSTRUCTIONS sheet.

Each operand address is encoded with the binary form of its decimal value. For example:



9. On the WORD BLOCK the angled corner is the upper left hand corner.

10. Each player must only use one color of WORD BLOCK.

11. A black peg = 1. A white peg = 0.

12. There must be a WORD BLOCK for each card in the program.

13. As soon as a player has encoded his program, he must remove the program cards and proceed with the WORD BLOCKS. He may, however, use the REPERTOIRE OF INSTRUCTIONS sheet for reference.

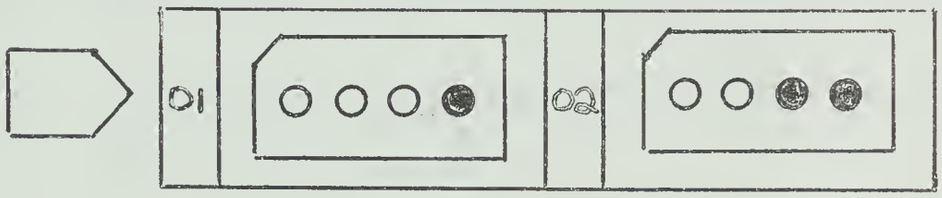
14. At this point each player must also prepare data words on WORD BLOCKS. These are the actual quantities to be input and processed by the computer. You need one WORD BLOCK for each variable in the program. You may select any value you wish for each variable, as long as it does not exceed 7.

15. Also, have ready blank WORD BLOCKS for values that result from computer operation. For example, for a program that solves a + b = c you must have WORD BLOCKS with values for a and b, and a blank WORD BLOCK for c.

16. At this point the player is ready to load his program into RANDOM ACCESS MEMORY of the computer. Each

player uses a terminal that is color coded the same as his WORD BLOCKS.

17. The first player to have a program loaded may start processing. He does this by placing his CONTROL ARROW in the CONTROL space opposite instruction register 01. At this time he proceeds to perform the instruction. For example:



reads "input a value to core memory location 3." The player will then place a data WORD BLOCK into CORE MEMORY location 3.

18. After performing one complete instruction, a player must say "next!" and the player on the right may start processing if his program is loaded. If it is not, play passes to the next player, etc.

19. If no player is ready, player one moves his control arrow down a space, and performs the next instruction. When he is finished, play again passes to the right, after player one says "next!".

20. Each player must use only those parts of CORE MEMORY and of the PROCESSOR BUFFER that are color coded the same as his program words.

21. If, at the end of an instruction, any information is left in the PROCESSOR, it must be stored in the PROCESSOR BUFFER until that player's next turn.

22. Any player who discovers any other player making an error in programming, encoding, or processing, gains an extra turn.

23. The player making the error loses a turn.

24. The player who gets his program through the computer first wins the game.

A P P E N D I X B

REPERTOIRES OF INSTRUCTIONS

PROGRAM INSTRUCTIONS FOR THE FIRST REVISION

PROGRAM INSTRUCTIONS

1	INP	input
2	OUT	output
3	ADD	add
4	SUB	subtract
5	LDB	load combine
6	LDP	load compare
7	JUP	jump
8	COJ	conditional jump
9	COM =	compare, equality
10	COM ≠	compare, not equal
11	INC	increment
12	STO	store
13	START	system command, start
14	STOP	system command, stop

a = 1

b = 2

c = 3

d = 4

e = 5

PROGRAM INSTRUCTIONS FOR THE SECOND REVISION

PROGRAM INSTRUCTIONS

1.	INP	input to core memory
2.	OUT	output from core memory
3.	LDP	load processor
4.	CLP	clear processor
5.	ADD	add to processor
6.	SUB	subtract from processor
7.	MPY	multiply
8.	DIV	divide
9.	INC	increment
10.	STO	store in core memory
11.	JUP	jump
12.	COJ	conditional jump (X = 1)
13.	STP	stop

1. a
2. b
3. c
4. d
5. e

PROGRAM INSTRUCTIONS FOR THE FINAL VERSION

REPERTOIRE OF INSTRUCTIONS

X - indicates an operand address for instructions 1 to 8. For instructions 9 to 11, X indicates the value to be used in the program.

1.	INP	place a quantity in X.
2.	OUT	transfer contents of X to output register.
3.	LDP	clear processor, and transfer the quantity in X to the processor.
4.	ADD	add contents of X to processor.
5.	SUB	subtract contents of X from processor.
6.	MPY	multiply contents of processor by contents of X.
7.	DIV	divide contents of processor by contents of X.
8.	STP	store the contents of processor in X.
9.	JUP	jump to program instruction X.
10.	COJ	jump to program instruction X if processor is negative.

11.	HLT	stop. The operand must be a row of zero's to maintain parity.
12.	INC	increment the processor by X.

A P P E N D I X C

TEST INSTRUMENTS

KNOWLEDGE OF CONCEPTS TEST

NAME_____

Questions about the game Timesharing, and computers.

1. Which two units in the game are shared by all players?

   a. _____

   b. _____

2. A computer has five main parts. What are these?

   a. _____

   b. _____

   c. _____

   d. _____

   e. _____

3. Write the binary number for 4.

4. What is the decimal number for this binary number?

   101 binary = _____ decimal

5. In the game, how does information get to the output unit?

   _____

   _____

6. Why does a computer have a special output unit?

   _____

   _____

7. The binary coded numbers for

INP
3

are 0001 and 0011

Encode this on the following two word blocks.

8. How do you think an I.B.M. card contains information?

_____

_____

9. In the game, how do you know which section of core memory you may use?

_____

_____

10. Can you explain how information might be stored in the memory of a real computer?

_____

_____

11. When playing the game, when do you input information into the computer?

_____

_____

12. How would information on I.B.M. cards get into a computer?

_____

_____

13. If, while playing the game, you were adding the two numbers 0011 and 0001,

    a. where would they be added?

    b. what would the answer be expressed as a binary number?

    a. _____

    b. _____

14. What is a processor buffer, and why is it necessary?

_____

_____

15. A computer instruction in the game might read

INP

5

What exactly does this mean?

_____

_____

16. What is a computer program?

_____

_____

17. What happens in the control section of the game?

_____

_____

18. What do you think the control section in a real computer does?

_____

_____

ATTITUDE QUESTIONNAIRE

Place a checkmark on the space corresponding to the description which best describes the way you feel about each of the following statements.

	strongly disagree	disagree	undecided	agree	strongly agree
1. Timesharing is fun. _____	( )	( )	( )	( )	( )
2. Timesharing is more fun than Krypto.	( )	( )	( )	( )	( )
3. Timesharing is a bore. _____	( )	( )	( )	( )	( )
4. Timesharing is too complicated. _____	( )	( )	( )	( )	( )
5. I would like to play Timesharing with tougher problems. ___	( )	( )	( )	( )	( )
6. I would rather do regular schoolwork than play Timesharing. _____	( )	( )	( )	( )	( )
7. I would like to have another chance to play Timesharing in school. _____	( )	( )	( )	( )	( )
8. Timesharing is too confusing.	( )	( )	( )	( )	( )
9. I would like to have a Timesharing game of my own. _____	( )	( )	( )	( )	( )
10. I don't think my friends would like Timesharing. _____	( )	( )	( )	( )	( )

If you have any comments or statements to make about the game, you may write them here.

_____

_____

_____

_____